

REMARKS

Applicant respectfully requests that the above-identified patent application be reexamined and reconsidered.

I. Introduction

Claims 1-18 are pending in the present application. In a September 21, 2004, Office Action (hereinafter "Office Action"), Claims 15-18 were objected to as being in improper dependent form. Claims 1-14 were rejected under 35 U.S.C. § 101 as being directed to unpatentable subject matter. Also Claims 1, 2, 15, and 16 were rejected under 35 U.S.C. § 102(e) as being anticipated in view of the teachings of U.S. Patent No. 6,675,230 to De Armas et al. (hereinafter "De Armas"). Claims 3-14, 17, and 18 were rejected under 35 U.S.C. § 103(a) as unpatentable over De Armas in further view of the teachings of U.S. Patent No. 6,675,230 to Lewallen et al. (hereinafter "Lewallen"). Pursuant to 37 C.F.R. § 1.111 and for the reasons set forth below, applicant respectfully requests reconsideration and allowance of the present application.

For the following reasons, applicant respectfully submits that Claims 1, 2, 15, and 16 are not anticipated by De Armas. Further, applicant respectfully submits that Claims 3-14, 17, and 18 are not obvious over De Armas in view of Lewallen. As will be explained in greater detail below, the cited and applied references, alone or in combination, fail to teach or suggest a system and method for adapting and hosting legacy user interface controls in a new window manager. Prior to discussing more detailed reasons for applicant's belief that all the claims of the present application are allowable, descriptions of the present invention and the cited references are presented. The following descriptions of applicant's invention and the cited references are not provided to define the scope or interpretation of any of the claims of this application.

LAW OFFICES OF
CHRISTENSEN O'CONNOR JOHNSON KINDNESS^{PLLC}
1420 Fifth Avenue
Suite 2800
Seattle, Washington 98101
206.682.8100

Instead, these descriptions are provided to help the United States Patent and Trademark Office better appreciate important claim distinctions discussed thereafter.

A. Summary of the Present Invention

The present invention is directed to a method and apparatus for adapting and hosting legacy user interface controls in a new window manager. Aspects of the invention may be embodied in a computer operating system capable of displaying a graphical user interface. Generally described, the present invention provides a bridge between a hosted legacy user interface control and a legacy window manager. The bridge intercepts and filters messages intended for the hosted user interface control to determine if the messages should be passed to the new window manager. If a message is to be passed, in one exemplary embodiment of the invention, the message is forwarded to the root Visual Gadget in the new window manager. The message is processed and routed down the window tree to an adapter control for hosting the legacy user interface control. The adapter control processes the message and routes the message to any listener objects attached to the adapter.

The present invention exposes legacy user interface controls, such as window handles, as native user interface controls to a new window manager. In order to expose legacy user interface controls as native interface controls in a new windows manager, routines of the present invention "hook" into the messaging infrastructure of the legacy window manager and intercept messages intended for the legacy user interface. The present invention supports the use of existing user interface controls in a new window manager without requiring source modification of the controls. The invention also allows existing controls to be identified as being "adapted," without requiring special handling.

LAW OFFICES OF
CHRISTENSEN O'CONNOR JOHNSON KINDNESS^{PLLC}
1420 Fifth Avenue
Suite 2800
Seattle, Washington 98101
206.682.8100

B. U.S. Patent. No. 6,675,230 to De Armas et al.

De Armas is directed to a method of adding functionality to an existing application program that utilizes a user interface. Additional processing capabilities are added to the target application along with additional user interface features by monitoring all of the target program's inputs and outputs. In order to monitor the activities of the target application and add the additional processing capabilities, a surrogate window procedure contained in a Dynamic Link Library ("DLL") is inserted into the process address space of the target application program. The surrogate window procedure initiates a "sub-classing" process in which a pointer to the target application's main window procedure is overwritten with a new pointer to the surrogate window procedure. This "subclassing" process results in all future messages to the target application being dispatched to the surrogate window procedure. Also, the "subclassing" process gains control over the memory used by the target application.

C. U.S. Patent. No. 6,675,230 to Lewallen et al.

Lewallen purports to disclose a system, method, and program for implementing components of a user interface as an object. More specifically, Lewallen is directed to allowing a Java-based application program to manipulate objects and interfaces in a user interface program such as Internet Explorer that is implemented using a standard Application Programming Interface ("API"). By allowing a Java-based application program to manipulate objects of a native user interface program (i.e., Internet Explorer), the Java-based program may implement its user interface using the native interface components. As a result, the Java-based program may be displayed with the same "look and feel" as the native user interface program.

II. Claim Objections

The Office Action objected to Claims 15-18 as being in improper dependent form. To overcome the objection, Claims 15-18 are canceled. Claims 19-27, which include one

independent claim and dependent claims that limit the subject matter of the independent claim, were added. Stated differently, some of the canceled claims have been rewritten in independent form to overcome the Examiner's objection.

III. Non-Statutory Subject Matter Rejection Under 35 U.S.C. § 101

The Office Action rejected Claims 1-14 under 35 U.S.C. § 102 as being directed to nonstatutory subject matter that could be practiced mentally in conjunction with pen and paper. Applicant has adopted the Examiner's suggestion to replace "method" with a "computer-implemented method" in the claims in order to overcome this rejection.

IV. The Claims Distinguished

A. Rejection of Claims 1-2, Under 35 U.S.C. § 102(e)

The Office Action rejected Claims 1 and 2 under 35 U.S.C. § 102(e) as being anticipated by De Armas. The Office Action asserts that De Armas discloses each and every element of these claims. As described in more detail below, applicant respectfully disagrees, as the cited reference fails to disclose or suggest certain elements of independent Claim 1 and dependent Claim 2.

1. Claim 1

Claim 1, as amended, recites:

1. A computer implemented method for hosting a legacy user interface object originally intended for use in a windows-based legacy window manager in a windows-based new manager, comprising:

providing a software bridge between said legacy window manager and said legacy user interface object that allows said legacy user interface object to operate in a computer that includes said new window manager and said legacy window manager;

intercepting a message at said software bridge intended for said legacy user interface object;

LAW OFFICES OF
CHRISTENSEN O'CONNOR JOHNSON KINDNESS^{PLLC}
1420 Fifth Avenue
Suite 2800
Seattle, Washington 98101
206.682.8100

determining whether said message should be forwarded to said legacy window manager; and

in response to determining that said message should be forwarded, forwarding said message to said legacy window manager.

Claim 1 defines a method where a software bridge is provided between specific types of software modules. More specifically, the claimed method recites "providing a software bridge between said legacy window manager and said legacy user interface object. . ." The method disclosed in De Armas does not provide a software bridge between a "*legacy window manager*" and a *legacy user interface object*" as recited in Claim 1. Instead, De Armas discloses a system where software routines are injected directly between "a computer operating system and the target program so as to act upon messages and commands to the target program." De Armas at Abstract. The target program is any program in which "modified user interface features or modified functionality" will be provided. *Id.* at Abstract. Simply stated, De Armas discloses a system for adding functionality to a graphically based "target program" in which the source code for the target program is unavailable to developers that want to provide the additional functionality. As stated in De Armas:

Likewise, it would be desirable for such third party developers to be able to seamlessly integrate the new function into a graphical user interface (GUI) of the existing application's user interface. In this regard, it should be understood that, as used herein, the phrase 'seamless integration' means that the user interface for the additional functionality to be added to a particular application, appears to a user to be an integrated or cohesive part of the existing software application's user interface, and not as a separate window or separate program.

De Armas at Col. 2, lines 1-10.

In order to add functionality to a target program, the De Armas system injects "itself directly between a computer operating system and the target program so as to intercept messages and commands to the target program. . ." De Armas at Col. 2, lines 50-53. As a result, "by

intercepting and then performing special processing of those messages to the target application which relate to the new functionality and/or which determines the appearance of the target application user interface," the De Armas system may add "new functional features and seize control over the appearance of the target application user interface." De Armas at Col. 2, lines 53-57.

Similar to the De Armas system, the present invention acts as an intermediary between two software modules. However, the present invention is not directed to adding additional functionality to an existing program. Instead, as recited in Claim 1, the present invention provides a software bridge between a "*legacy window manager*" and a "*legacy user interface object*." As known to those skilled in the art and others, developers are constantly improving software systems. One problem with building a new software system is that old or legacy program code designed to work in conjunction with a software system being replaced will not normally be compatible with the new system. With respect to the present invention, graphical user interfaces "typically employ some form of a window manager to organize and render windows." Application at page 1. When creating an entirely new window manager, objects designed to work with the legacy window manager would not normally be compatible with the new window manager. However, the present invention provides a software bridge "between said legacy window manager and said legacy user interface object." Then, messages are intercepted at the software bridge and routed to either the new or legacy window manager, depending on whether the message is from a new or legacy user interface object.

The Office Action states that De Armas provides a software bridge between a legacy window manager and a legacy user interface object. In support of that proposition, the Office Action cites the following sections of De Armas: Figure 3; Col. 4, lines 1-38; and Col. 6, lines 25-50. However, the referenced sections of De Armas do not disclose a software bridge

LAW OFFICES OF
CHRISTENSEN O'CONNOR JOHNSON KINDNESS^{PLLC}
1420 Fifth Avenue
Suite 2800
Seattle, Washington 98101
206.682.8100

between a legacy window manager and a legacy user interface object. Instead, the Office Action incorrectly equates a legacy window manager as disclosed in the present invention with a window procedure as disclosed in De Armas. Specifically, De Armas states: the "code associated with the window is referred to as a window procedure. Generally speaking, the window procedure can be located within the application program or in a dynamic link library (DLL) . . ." Also, a "window procedure has two primary functions: (1) it defines what is to be displayed within a particular window's boundaries or client area on a user interface screen . . . and (2) determines the precise manner in which the window will respond to certain user inputs." De Armas at Col. 4, lines 5-18. A legacy window manager is not equivalent to a window procedure. As stated in the present application, a window manager organizes and renders a plurality of windows for different applications by utilizing "a window tree to organize windows, their child windows, and other user interface controls to be displayed within the window such as buttons, menus, etc." Application at page 1, lines 14-16. A window procedure as disclosed in De Armas is program code used by an application to communicate with an operating system. By contrast, a window manager is a component of an operating system that manages windows created by one or more applications. Thus, the Office Action incorrectly equates a window procedure with a legacy window manager. Since De Armas does not disclose a system for providing a software bridge between a legacy window manager and a legacy user interface object, some of the claim elements recited in Claim 1 are not disclosed in De Armas.

Claim 1 further defines a method where a legacy user interface object is adapted for use in a computer that includes specific components. More specifically, the claimed method recites "providing a software bridge between said legacy window manager and said legacy user interface object that allows said legacy user interface object **to operate in a computer that includes said new window manager and said legacy window manager.**" As described

LAW OFFICES OF
CHRISTENSEN O'CONNOR JOHNSON KINDNESS^{PLC}
1420 Fifth Avenue
Suite 2800
Seattle, Washington 98101
206.682.8100

previously, De Armas discloses a system where software routines are injected between a computer operating system and the target program in order to add functionality to the target program. However, applicant is unable to find any reference in De Armas to a system for allowing a legacy user interface object to operate in a computer that includes a new and legacy window manager. Simply stated, De Armas is directed to adding new functionality to an existing program and does not disclose any techniques for hosting legacy user interface objects in a new window manager.

Claim 1 further defines a decision-making process that includes "determining whether the message should be forwarded to said legacy window manager." The Office Action states that De Armas discloses the same decision-making process and cites De Armas at Col. 5, lines 25-49, in support of that proposition. However, the referenced section of De Armas does not disclose the same decision-making process. More specifically, De Armas states:

the technology injection system (TIS) 106, intercepts all messages (queued and non-queued) from the operating system 102, **to the target application program 100**. More particularly, upon receiving a message from the operating system 102, the TIS 106 evaluates the message to determine whether it concerns a function or GUI feature **which is to be modified** or implemented by the TIS program.

(Emphasis added). De Armas at Col. 5, lines 28-33.

The present invention determines whether a message should be forwarded to a legacy window manager. By contrast, the referenced section of De Armas describes intercepting a message and determining whether the message is related to functionality "which is to be modified" by the De Armas system. If a message is related to new functionality and needs to be modified, the message is forwarded to program code designed to make the necessary modification (e.g., the TIS program). In order to make these modifications, messages from the operating system to "the target application program" are intercepted. By contrast, the present

LAW OFFICES OF
CHRISTENSEN O'CONNOR JOHNSON KINDNESS^{PLC}
1420 Fifth Avenue
Suite 2800
Seattle, Washington 98101
206.682.8100

invention determines whether a message "should be forwarded to said legacy window manager." As known to those skilled in the art and others, a window manager is a component of an operating system that manages windows created by one or more applications. Simply stated, the decision-making process involved in determining if a message should be forwarded to a new or legacy window manager is not the same as determining if a message is related to functionality that will be modified. Thus, the recited claim element of "determining whether the message should be forwarded to said legacy window manager" is not disclosed by De Armas.

To establish a proper rejection under 35 U.S.C. § 102, M.P.E.P. § 2143 states that "[a] claim is anticipated only if each and every element as set forth in the claim is found, either expressly or inherently described, in a single prior art reference." M.P.E.P. § 2131 (February 2003). The M.P.E.P. further states that "[t]he identical invention must be shown in as complete detail as is contained in the . . . claim." *Id.* Applicant respectfully submits that De Armas fails to expressly or inherently teach, disclose, or suggest each and every element of Claim 1. For instance, as explained above, De Armas fails to disclose or suggest "providing a software bridge between said legacy window manager and said legacy user interface object that allows said legacy user interface object to operate in a computer that includes said new window manager and said legacy window manager." Accordingly, for at least these reasons, applicant respectfully submits that the rejection of Claim 1 is in error and request that it be withdrawn.

2. Claim 2

Since Claim 2 depends from Claim 1, the analysis applied to Claim 1 also applies to Claim 2. Therefore, applicant respectfully submits that Claim 2 is in condition for allowance for the same reasons as Claim 1. In addition, applicant submits that Claim 2 is allowable for additional reasons described below.

LAW OFFICES OF
CHRISTENSEN O'CONNOR JOHNSON KINDNESS^{PLLC}
1420 Fifth Avenue
Suite 2800
Seattle, Washington 98101
206.682.8100

Claim 2 is directed to instances in which a message is not intended for the legacy window manager. More specifically, Claim 2 recites:

2. The method of Claim 1, further comprising:

in response to determining that said message should not be forwarded to said legacy window manager, forwarding said message to a procedure originally intended to handle said message.

Applicant respectfully submits that "determining that said message should not be forwarded to the legacy window manager" is not equivalent to passing a message to a target window procedure when the message does "not concern any new functionality or display of the window elements comprising the target application GUI interface." De Armas at Col. 10, lines 35-38. As mentioned above, the Office Action incorrectly equates a legacy window manager as disclosed in the present invention with a window procedure as disclosed in De Armas. Therefore, since De Armas fails to expressly or inherently teach, disclose, or suggest each and every element of Claim 2, applicant respectfully submits that the rejection of Claim 2 is in error and request that it be withdrawn.

B. Rejection of Claims 3-5, Under 35 U.S.C. § 103(a)

The Office Action rejected Claims 3-5 under 35 U.S.C. § 103(a) as being unpatentable over De Armas in view of Lewallen. The Office Action asserts that De Armas and Lewallen suggest each and every element of these claims and that it would be obvious to combine the teachings of De Armas and Lewallen. Applicant respectfully disagrees.

Since Claims 3-5 depend from Claim 1, the analysis applied to Claim 1 also applies to Claims 3-5. Therefore, applicant respectfully submits that Claims 3-5 are in condition for allowance for the same reasons as Claim 1. In addition, applicant submits that Claims 3-5 are allowable for additional reasons described below.

LAW OFFICES OF
CHRISTENSEN O'CONNOR JOHNSON KINDNESS^{PLLC}
1420 Fifth Avenue
Suite 2800
Seattle, Washington 98101
206.682.8100

With respect to Claim 3, applicant submits that De Armas, alone or in combination with Lewallen, fails to disclose or suggest the claimed combination of steps, including "forwarding said message to a root user interface object hosted in a window tree maintained by said new window manger." The method disclosed in Lewallen does not disclose forwarding a message to a root user interface object hosted in a window tree maintained by a new window manager as recited in Claim 3. Instead, Lewallen discloses a system where native user interface objects are made accessible to a Java Program. As a result, the Java program can implement its user interface using the native interface component objects of the user interface program. "This allows a Java program user interface to have the same 'look-and-feel' as the user interface program, e.g., Internet Web browser, as the Java program is ultimately using the same user interface component objects utilized by the user interface program." Lewallen at Col. 4, lines 10-14. In order to give Java programs the same look-and-feel as a native program, the "root user interface (UI) document object and Java document object are obtained ..." and manipulated. Lewallen at Col. 13, lines 32-24. The Office Action incorrectly equates a "root user interface object **hosted in a window tree**" as disclosed in the present invention with a "root user interface **document object**," as disclosed in Lewallen. As known to those skilled in the art and others, a document object is a representation of a document as a hierarchical arrangement of nodes which contains a root node. All of the data needed to display a document is contained in the document object. The Document Object Model ("DOM") is a standardized format for arranging elements of a document in a tree data structure. A set of APIs is available from the DOM specification that allow developers "to access, change, delete, or add nodes to the DOM representation of a document, or application program." Lewallen at Col. 5, lines 23-25. Document objects are merely processed by rendering engines when displaying documents and are not "hosted in a window tree." By contrast a root user interface object, as disclosed in the

LAW OFFICES OF
CHRISTENSEN O'CONNOR JOHNSON KINDNESS^{PLC}
1420 Fifth Avenue
Suite 2800
Seattle, Washington 98101
206.682.8100

present invention, represents a window, button, or other graphically based user interface control. Graphical objects are hosted in a window tree so that a window manager may keep "track of visible windows and other user interface controls, and render the controls in video memory." Present application at page 7, lines 17-20. Simply stated, a "root user interface document object," as disclosed in Lewallen is not equivalent to a root interface object that is "hosted in a window tree" as recited in Claim 3. Thus, for at least these reasons, applicant respectfully submits that the rejection of Claim 3 is in error and request that it be withdrawn.

Dependent Claim 4 adds to the nonobviousness of applicant's invention, including the element of "routing said message down said window tree maintained by said new window manager to an adapter control associated with said legacy user interface object." The Office Action asserts that De Armas as modified by Lewallen teaches routing messages down a window tree maintained by a window manager. As discussed above, Lewallen manipulates tree data structures that satisfy the DOM specification. However, applicant is unable to find any reference in Lewallen to routing a message down a window tree as recited in Claim 4. Accordingly, the cited references fail to teach or suggest the additional elements recited in Claim 4.

Dependent Claim 5 adds to the nonobviousness of applicant's invention, including the element of "processing said message at said adapter control operative to convert said message into a format understood by said legacy window manager." The Office Action asserts that De Armas as modified by Lewallen teaches processing a message at an adapter control and cites Col. 11, lines 30-50 of Lewallen in support of that proposition. The cited section of Lewallen states: "Each Kit 202a, further includes a stub factory class 208a, that provides methods to handle W3C API calls to objects intended for the user interface program." Lewallen at Col. 11, lines 46-48. However, a stub factory class as disclosed in Lewallen is not equivalent to an adapter control as disclosed in the present invention. More specifically, an adapter control

LAW OFFICES OF
CHRISTENSEN O'CONNOR JOHNSON KINDNESS^{PLLC}
1420 Fifth Avenue
Suite 2800
Seattle, Washington 98101
206.682.8100

converts a "message into a format understood by said legacy window manager." Conversely, a stub factory class handles "W3C API calls to objects intended for the user interface program." Accordingly, the cited references fail to teach or suggest the additional elements recited in Claim 5.

C. Rejection of Claim 10 Under 35 U.S.C. § 103(a)

The Office Action rejected Claim 10 under 35 U.S.C. § 103(a) as being unpatentable over De Armas in view of Lewallen. The Office Action asserts that De Armas and Lewallen suggest each and every element of this claim and that it would be obvious to combine the teachings of De Armas and Lewallen. Applicant respectfully disagrees. Since Claim 10 contains language that parallels Claims 1, and 3-5, the analysis applied to these claims also applies to Claim 10. Therefore, applicant respectfully submits that Claim 10 is in condition for allowance for the same reasons as Claims 1, and 3-5, provided above.

D. Rejection of Claims 6-9 and 11-14, Under 35 U.S.C. § 103(a)

The Office Action rejected Claims 6-9 and 11-14 under 35 U.S.C. § 103(a) as being unpatentable over De Armas in view of Lewallen. The Office Action asserts that De Armas and Lewallen suggest each and every element of these claims and that it would be obvious to combine the teachings of De Armas and Lewallen. Applicant respectfully disagrees. Since Claims 6-9 depend from Claim 1 and Claims 11-14 depend from Claim 10, the analysis applied to Claims 1 and 10 provided above also applies to these claims. Therefore, applicant respectfully submits that Claims 6-9 and 11-14 are in condition for allowance for the same reasons as Claims 1 and 10, respectively. In addition, applicant submits that Claims 6-9 and 11-14 are allowable for additional reasons described below.

The Office Action asserts that De Armas as modified by Lewallen teaches "forwarding a message to a procedure originally intended to handle said message from said adapter control," as

LAW OFFICES OF
CHRISTENSEN O'CONNOR JOHNSON KINDNESS^{PLLC}
1420 Fifth Avenue
Suite 2800
Seattle, Washington 98101
206.682.8100

recited in Claims 6 and 11. However, applicant is unable to find any reference in Lewallen to an adapter control. Moreover, Lewallen does not forward messages to procedures originally intended to handle the message. Instead, Lewallen manipulates data in a DOM tree in order to create a native user interface look-and-feel in Java programs. Accordingly, the cited references fail to teach or suggest the additional elements recited in Claims 6 and 11.

Dependent Claims 7 and 12 add to the nonobviousness of applicant's invention, including the element of "routing said message from said adapter control to a listener object attached to said adapter control." The Office Action asserts that De Armas as modified by Lewallen teaches forwarding a message to a listener object and cites Col. 10, lines 43-46 of Lewallen in support of that proposition. The referenced sections of Lewallen state that the "mixed statement program could include event listeners to modify the HTML page." Lewallen at Col. 10, lines 43-46. Applicant submits that "routing" a message to a listener object is not equivalent to merely including a listener object on an HTML page. Accordingly, the cited references fail to teach or suggest the additional elements recited in Claims 7 and 12.

Dependent Claims 8 and 13 add to the nonobviousness of applicant's invention, including the element of "determining whether said message has been completely handled." The Office Action asserts that De Armas as modified by Lewallen teaches determining whether a message has been completely handled. However, as described previously, handling a message that is received at a window manager, as disclosed in the present invention, is not equivalent to handling a message that relates to adding functionality to a target application as disclosed in Lewallen. Accordingly, the cited references fail to teach or suggest the additional elements recited in Claims 8 and 13.

LAW OFFICES OF
CHRISTENSEN O'CONNOR JOHNSON KINDNESS^{PLLC}
1420 Fifth Avenue
Suite 2800
Seattle, Washington 98101
206.682.8100

CONCLUSION

In view of the foregoing remarks, it is submitted that the present application is now in condition for allowance. Reconsideration and reexamination of the application and allowance of the claims are solicited. If the Examiner has any questions or comments concerning this matter, the Examiner is invited to contact applicant's undersigned attorney at the number below.

Respectfully submitted,

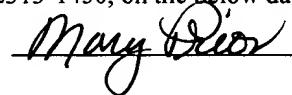
CHRISTENSEN O'CONNOR
JOHNSON KINDNESS^{PLLC}



Clint J. Feekes
Registration No. 51,670
Direct Dial No. 206.695.1633

I hereby certify that this correspondence is being deposited with the U.S. Postal Service in a sealed envelope as first class mail with postage thereon fully prepaid and addressed to MAIL STOP AMENDMENT, Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450, on the below date.

Date: January 20, 2005



CJF:mgp

LAW OFFICES OF
CHRISTENSEN O'CONNOR JOHNSON KINDNESS^{PLLC}
1420 Fifth Avenue
Suite 2800
Seattle, Washington 98101
206.682.8100